Abstract—This paper details a full-text search interface developed to support mission operators to quickly search across multiple databases simultaneously and allow for search result refinement using a faceted user interface. By leveraging the contextualized NETMARK[1] eXtensible Database (XDB), this search interface automatically inferences categories for facets and allows users to select one or more category for each facet to narrow the results to only display those that meet all of the selected facet categories.

TABLE OF CONTENTS
1. INTRODUCTION ........................................1
2. APPROACH ............................................5
3. IMPLEMENTATION DETAILS ..........................6
4. RESULTS .............................................6
4. CONCLUSION .........................................7
5. BIOGRAPHIES .........................................8
6. REFERENCES .........................................8

1. INTRODUCTION

The immense success of commercial search systems such as those deployed at Google, Amazon, eBay, and other top-tier search and commerce sites illustrate the demand from end-users for convenient and expedient access to the relevant information or products the user seeks. Naive and sophisticated users of search systems today demand a simple, easy-to-use search mechanism within which they can type in words, identifiers, or phrases of interest and the results are presented with the least amount of effort by the user. These users expect relevant results and the ability to iteratively fine-tune their results, usually by adding more terms to the search phrase.

Search Context

Generalized search systems for searching multiple Internet or Intranet sites are generally limited in their ability to control the “context” of their search based upon fragments of the URL addressing space, such as host name and path of the pages indexed. This capability is particularly powerful for users familiar with a subset of web sites that contain the information most relevant to them and most likely to address the questions asked through search. Additional “document contexts” are often available, depending on the types of systems indexed. For example, map systems can be searched on geographical or data set contexts, message boards can be searched on author, date, or other relevant metadata.

For search systems that do not have a-priori knowledge of the structure and metadata of the data they are indexing, they are unable to retrieve results based on more refined contexts. For example, searching the Internet for “Christopher D. Knight” will return a large number of results, but if the user was interested in papers on which Mr. Knight were the author, and the user was able to limit the search to the “Authors” section of documents, it would significantly narrow the number of results to a more manageable and relevant set. Leveraging additional contexts, such as site (say, IEEE sites) would increasingly target the results for further accuracy.

Incremental Refinement of Search

While sophisticated users may already have preconceived contexts for their searches, most users find complex grammars or user interfaces to be a barrier to use of the search system. Instead, this paper proposes an incremental refinement approach wherein the user performs a simple keyword search across all contexts and, post search, refines the results to narrow them to a smaller relevant set.

Incremental refinement has additional benefits—the user may want to explore the larger space of results and may find results that are relevant but not necessarily in the contexts the user expected. Or, the user may not know the context(s) until they perform the initial search. Sophisticated experienced users, if given the choice to narrow the search before exploring, may miss relevant results. Using the previous example, if a sophisticated user were able to limit the search to the Authors section of IEEE papers, that user would not be aware of the instances where Mr. Knight appears in the references section of the papers, or that Mr. Knight has a patent with the USPTO on a technology relevant to the user’s interests.

11-4244-1488-1/08/$25.00 ©2008 IEEE
2 IEEEAC paper #1091. Version 2, Updated 2007.10.20
Many e-commerce sites not only provide for keyword search of their product database with incremental (and increasingly-complex) search criteria, but also for post-search refinement of the results based on multiple “facets” [2]. Facets are classifications for indexing and information retrieval that break out of the strict hierarchy of a single taxonomy, to represent multiple category views of objects. Multi-faceted search presents users categories in multiple dimensions to choose from for searching or refining a search [3].

Usually these facets leverage one or more taxonomies (also used for browsing for products) developed and maintained by the commerce site. They are often associated with thesauruses for text-based indexing. Even though facets are most often defined manually, there are several ways to dynamically generate facet structures through text analysis, data mining, and document structure analysis, and other techniques. There is considerable research in automatically generating these taxonomies given a set of input data. The Flamenco project[4] has been the subject of both automatic generation of facet categories given a corpus of input data and user interfaces for search with integrated faceted navigation and browsing.

For International Space Station (ISS) mission operations, software problem tracking, change tracking, and related documents are stored in the PVCS problem report database system. PVCS stands for the vendor name for the ISS Change Paper and Version Control System. The PVCS system is used to research ISS flight software change requests and change documents. Other sets of documents including procedures, handbooks, and the like are constantly maintained by operators when they are not on console or in training, and this process of updating requires constant awareness of the state of systems including referencing systems such as PVCS.

The norm for displaying facets on e-commerce sites is to show an ordered list of facets such as illustrated in the layout shown in Figure 1 (with the facet containing the most number of matches first) in a tight column on the left side of the page and the top matches on the right. Each facet is listed with the number of results in the total number of results that belong to that facet.

For example, on Amazon.com, searching for “Batman” returns a set of results in the main section of the window and the left-hand column lists “Books (14,759), Toys & Games (1,588)...” Clicking on one of these categories narrows the set of results to only those in the selected category. For example, clicking on Books will focus on only the books matching the search term of Batman. Other e-commerce sites add additional facets such as price ranges, years of publication, or seller location where the user can narrow the results based on categories within these additional facets.
As illustrated in Figure 2, the existing PVCS web-based search interface is time-consuming to use and restrictive—users seeking all records regarding a particular ISS subsystem or component and must perform a large number of search operations to answer a simple question such as, “What problems have been reported and resolved regarding GPS software?” Searching each set of PVCS records requires multiple user operations in a complex query interface. This search interface requires selecting one database such as SCR and one or at most two fields such as the “title” or “owner”. All records are presented as HTML tables for viewing, with references to related records provided in the HTML tables as links.

**NETMARK Search System**

Developed by a team in the Intelligent Systems Division of the NASA Ames Research Center, the NETMARK system provides an environment for data integration across multiple, disparate, and heterogeneous repositories. NETMARK combines the capability to ingest a variety of data in unstructured, semi-structured, and highly-structured forms with the ability to search the NETMARK index and retrieve information atoms via a rich, while straightforward query language. Data is integrated in NETMARK eXtensible Database (NETMARK XDB) with no a-priori knowledge of the structure and with no NETMARK configuration required. It is common, however, to massage the data being loaded into NETMARK for increased precision of retrievals. For example, unstructured text, Word documents, HTML documents, and XML documents may be loaded into NETMARK XDB directly. However, semantic information about the relationship between sections of text is often not explicit in less-structured formats such as HTML (i.e. in HTML tables). As part of the NETMARK deployment, Ames developed a web crawling system that automatically retrieves updated PVCS records via the web interface, normalizes the table-heavy HTML into an XML structure, and loads the XML into the NETMARK database.

Data is loaded into NETMARK XDB via the Web Distributed Authoring and Versioning (WebDAV[5]) protocol, a file-sharing protocol built on top of the HyperText Transport Protocol (HTTP). As files are “transferred” to the NETMARK XDB system, they are automatically decomposed and indexed with a full-text index system and immediately available for querying and document fragment retrieval by clients and users.

Interacting with the NETMARK schema-less database system is performed by executing simple or compound searches using a specific grammar of “context + content” requests, i.e. searching for the content “Knight” under a context with “Authors” will produce a result set containing this and other papers where the name Knight appears in the Authors section of those documents. Searches in NETMARK XDB can be as simple as a “content-only” search (returns any context with the terms sought in the content section) or compound with multiple contexts and contents.
Figure 3. above, illustrates the unique nature of the NETMARK XDB, in that text fragments are returned by the query in a combined query result document. Re-composition of documents (i.e. taking all of the abstracts from a set of documents and producing a summary document for further distribution) is a straightforward and powerful capability provided by NETMARK and has been leveraged in a variety of applications.

Table 1. Example NETMARK Query + Response

<table>
<thead>
<tr>
<th>Request:</th>
<th><a href="http://xdb/xdbq/Authors/Knight">http://xdb/xdbq/Authors/Knight</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Response:</td>
<td><code>&lt;xml&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;resultset&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;result&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;uri&gt;/xdb/doc1.xml&lt;/uri&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;Authors&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;name&gt;Knight&lt;/name&gt;</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;uri&gt;/xdb/doc2.xml&lt;/uri&gt;</code></td>
</tr>
</tbody>
</table>

NETMARK queries not only return the records that match the search criteria but also return the context and all contents under that context that match. For example, the query in Table 1 returns the complete “Authors” section wherein the name Knight appears, not just the Uniform Resource Identifier (URI, often a Uniform Resource Locator [URL] but not necessarily [6]) of the record or a text snippet. The resulting XML is a structured set of data, for example it is common to return an XML sub-tree in the body of each result. That result structure allows for further analysis and formatting for presentation.

Often a client user-interface is provided that interacts with NETMARK XDB by using XSLT [7] (server or client-side), using Asynchronous JavaScript and XML (AJAX [8]), or any other client capable of making HTTP requests and processing XML responses such as Microsoft Excel.

Mission Operations Integrated Search

A team of developers at NASA Ames Research Center has been working to address PVCS search needs—with the goal of finding relevant documents quickly and easily with a minimum of user interactions.

The NETMARK XDB system was leveraged to index a subset of the PVCS records of particular interest to mission operators and provide unified search. The records extracted from PVCS are Software Change Requests (SCR), Station Program Notes (SPN), Change Notes (CN), and Schedule Issue/Change Forms (SIF).
The application of the NETMARK model to ISS mission operations has been an on-going project for many years and a variety of user interface prototypes have been developed. These previous interfaces required the user to select a context for their search, similar to the PVCS system, but more powerful in that the search operation would be performed across multiple record types simultaneously. For example, the previous system would allow users to select the “title” or “description” context using a pull-down form element and provide search terms in a text box. Documents containing the search terms in the context specified were returned. However, even sophisticated users were confused by the user interface when asked to both provide a search string and pre-select a context (in the case of data replicated from PVCS, record field name). Also of concern to the developers, while it would have been possible to allow the user to select “all contexts”, habit and familiarity with the PVCS search system would lead users to select the contexts they felt were most likely to return results, and records that may be of interest may be overlooked.

2. APPROACH

The first approach to addressing ease of use was to provide a simple user interface with a single text input field and perform a “content-only” search of the database with the words and phrases provided. Repeated requests from users to make the interface “as easy to use as Google” reinforced the need to simplify and provide the bare minimum required to perform the work. This simplified user interface produces a set of results in a paginated view for the client to browse through and to review individual results in detail. Links to the data record in the authoritative system of record are provided for each search result (akin to the links in common search service results such as Google) but an expandable display of results is also available for immediate viewing.

In developing this simplified user interface, it became apparent that the contexts of the resulting contents were available at no cost. By using the NETMARK XDB system, all searches produce the associated contexts for the contents found. The contexts of all of the results are compiled and presented to the end-user in a faceted user interface for further refinement. Each result is shown as one line of text

![Figure 4 – NX Search Facet UI](image)

but a small triangle button expands one record’s display, showing all of the fields recorded in NETMARK, for quick review. A link to the record in PVCS is also provided so that the system of record is consulted, and so that users may navigate to related records in a manner familiar to regular users of PVCS.

Users of this system are presented with a simple search form wherein they select the type(s) of records they wish to search and a free-text field for providing terms and phrases of interest. Screenshots of the user interface pages are shown in Figures 4 and 5.
Figure 4 illustrates the layout of the complete user interface, with the title, search field, and data set selection checkboxes at the top of the page, the faceted category selection interface on the left, and the results in the main section of the page.

![Figure 4 - Faceted User Interface](image)

**Figure 5 - NX Search Facet UI (Detail)**

Figure 5 provides an illustration of the faceted user interface, including the three main facets currently generated (record fields wherein the search terms were found, the record types wherein the terms were found, and the record status, as extracted from the records.) Each facet category is named (with lengthy names being summarized with ellipses, rolling over the shortened name shows a “tool tip” of the full name) and the number of records that are contained in that category. Each facet category is a clickable element that, when the user clicks the category that category is toggled on and off. Records in the main section of the display are only shown when it belongs to at least one category for each facet currently “on”. For example, for an SCR with “plasma” in the title and description, clicking the “title” category off will not remove that SCR, while clicking the SCR category under type or turning both title and description off under field will.

### 3. Implementation Details

The NETMARK search system includes a sophisticated system used to “crawl” PVCS on a regular basis and to convert records into a normalized XML form for insertion (via the WebDAV protocol) into the NETMARK XDB. Records are organized in NETMARK by category and state (PVCS fields) for increased control over the search “scope” and to provide additional search facets.

![NETMARK XDB Implementation Architecture](image)

**Figure 6 - NETMARK XDB Implementation Architecture**

Figure 6 illustrates this architecture for crawling, transforming, storing, and retrieving records.

Clicking the search button in the web user interface triggers an AJAX process that sends a query request to the server and presents the results in a paginated list form. The contexts of all of the results are compiled and presented to the end-user in a faceted user interface for further refinement. The AJAX code also collects and compiles the contexts provided in the results, as well as other metadata provided by the server, and produces a faceted user interface display. Further refinement is performed on the client, as the entire result set is delivered to the client in response to the content-only query submitted.

### 4. Results

Feedback from end-users has been extremely positive as the interface reduces the number of user interactions required to search the PVCS records and the user interface is generally, see below) intuitive and straightforward. As pagination, facet selection, and other user interactions are purely client-side, the interface performs well, even with a large number (thousands) of results. The AJAX implementation supports a wide variety of web browsers and generally presents the interface in a uniform way, but there remain minor differences in the ways elements are rendered.

This faceted user interface is novel in the following ways:

- Facet categories are automatically computed from the search results and do not require a pre-generated taxonomy of facets. This allows for integration of additional data sets at no cost, no taxonomy generation is required when new data sets are added to the database and search interface as categories are computed based on the results of the query. Further normalization of computed facets may be reasonably performed to ensure semantically equivalent categories are combined.
• Individual results may belong to more than one category under a given facet (e.g. if the user searched for “Knight” and it appeared in both the Author and Bio sections of a document, that document would belong to both categories. As facets and their categories are non-hierarchical, and as records may belong to more than one facet, the UI allows users to select multiple facet categories and they are presented with all records that belong to any of those facets. Other faceted UI’s rely on facet categories being hierarchical and only allow users to select an individual category under each facet.

• Interactions with the faceted UI are performed purely client-side for high performance. The authors are unaware of any faceted user interfaces that are performed purely on the client and that provide the immediate feedback and high performance inherent in this implementation.

The user interface has presented some challenges and have remaining, open questions; including:

• How to handle user selections -- When all categories for a facet are selected and the user selects one category, should all other categories be de-selected, or the one selected?

• What numerical value to display next to each facet category?—current implementation shows the total number of records in that facet category, but this value may be confusing when one facet has multiple categories turned off.

• How to handle very long facet category lists - currently the lists are unordered, but even when they are ordered, users may seek the categories with fewest results, or with the most results.

4. Conclusion

The NETMARK XDB Faceted Search user interface will continue to be improved based upon feedback provided by end-users. The techniques and layout will be leveraged for other applications at NASA and other agencies. It is applicable to systems that leverage the XDB server technology and semi-structured, contextualized nature of the query results produced. Further developments include evaluating how best to combine less-structured documents with more structured data sets such as the PVCS records. They also include leveraging ontologies and taxonomies to organize and unify the display in a more logical and user-friendly fashion. The technique may be easily applied to any search system that uses generated metadata to provide views of the information to be retrieved.
5. **Biographies**

**Christopher D. Knight** is a computer engineer at the NASA Ames Research Center. He is one of the key developers of the NETMARK XDB system. Before joining NASA, he worked for a number of Silicon Valley companies on web-based shopping, gaming, and collaborative systems. His interests are focused on asynchronous collaborative technologies and architectures for rapid application development tools for enabling collaboration. He received a bachelor's degree in computer science from Cal Poly, San Luis Obispo in 1996.

**Jane T. Malin** is Senior Technical Assistant in the Intelligent Systems Branch, Automation, Robotics and Simulation Division, Engineering Directorate, NASA Johnson Space Center, where she has led intelligent systems research and development since 1984. She has led research on model-based hazard analysis and simulation for engineering intelligent systems and on information extraction from text for modeling, using ontology-based semantic text analysis. Her 1973 Ph.D. in Experimental Psychology is from the University of Michigan, with a dissertation on mathematical modeling of information processing in problem solving.

6. **References**


